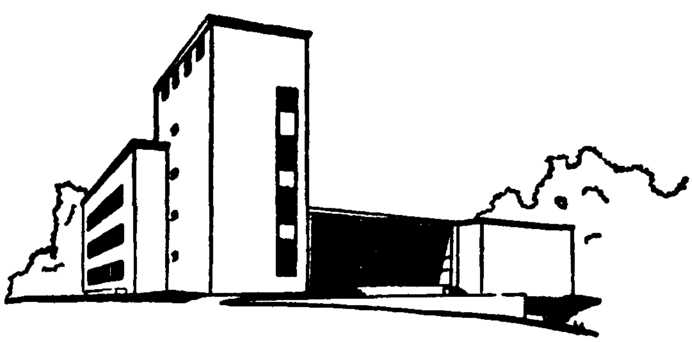
Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

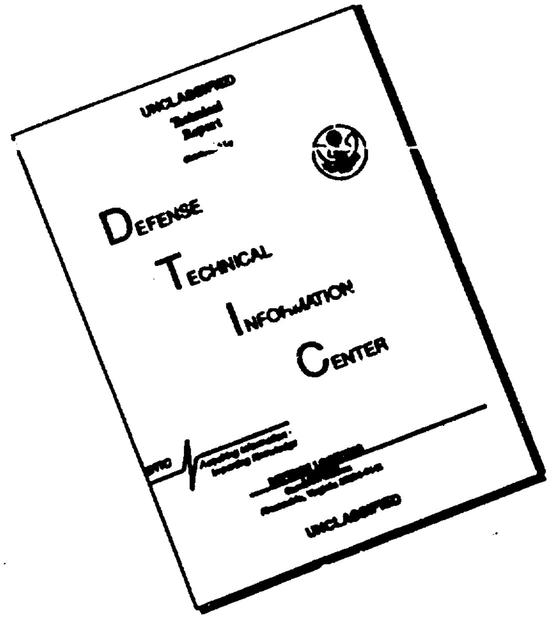
GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



36

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

W.P. # 103-70-1

Management Sciences Research Report No. 248

A SIMPLEX-LIKE ALGORITHM FOR THE

CONTINUOUS MODULAR DESIGN PROBLEM

by

Timothy L. Shaftel*
and
Gerald L. Thompson**

May 1971

Revised January 1972

* University of Arizona
** Carnegie-Mellon University

This report was prepared as part of the activities of the Management Sciences Research Group, Carnegie-Mellon University, under Contract N00014-67-A-0314-0007 NR 047-048 with the U. S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

Management Sciences Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Graduate School of Industrial Administration Carnegie-Mellon University	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP Not applicable

3. REPORT TITLE

A SIMPLEX-LIKE ALGORITHM FOR THE CONTINUOUS MODULAR DESIGN PROBLEM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Management Sciences Research Report May 1971

5. AUTHOR(S) (First name, middle initial, last name)

Timothy L. Shaftel
Gerald L. Thompson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1971	32	12

8a. CONTRACT OR GRANT NO. N00014-67-A-0314-0007 b. PROJECT NO. NR 047-048 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Management Sciences Research Report No. 248 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) W.P.# 103-70-1

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES Not applicable	12. SPONSORING MILITARY ACTIVITY Logistics and Mathematical Statistics Br. Office of Naval Research Washington, D. C. 20360

13. ABSTRACT

This paper derives an efficient solution procedure for solving the continuous version of the Evans modular design problem. The Kuhn-Tucker conditions are used to derive a dual problem which can be solved easily and whose dual variables indicate which equations should be tight. The technique retains a tree-basic solution throughout so that fast solution routines can be employed which are quite similar to those for transportation problems. Because of these analogies, the solution of transportation size problems can be effected with only moderately increased computer time.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

Unclassified
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Modular design						
Kuhn-Tucker						
Tree-basic						
Primal-dual						
Simplex-like						

DD FORM 1 NOV 65 1473 (BACK)

Unclassified
Security Classification

ABSTRACT

This paper derives an efficient solution procedure for solving the continuous version of the Evans modular design problem. The Kuhn-Tucker conditions are used to derive a dual problem which can be solved easily and whose dual variables indicate which equations should be tight. The technique retains a tree-basic solution throughout so that fast solution routines can be employed which are quite similar to those for transportation problems. Because of these analogies, the solution of transportation size problems can be effected with only moderately increased computer time.

1. INTRODUCTION

The modular design problem was first presented by David Evans [4]. In this problem, parts are to be grouped into a single module, several of which are then used in each application. The objective is to minimize the total cost of parts used:

$$\min_{x,y} \sum_{i \in I} c_i x_i' \sum_{j \in J} d_j y_j'$$

$$\left.\begin{aligned} \text{s.t.}\quad & x_i' y_j' \geq r_{ij}' \\ & x_i', y_j', c_i, d_j, r_{ij}' \geq 0 \\ & x_i', y_j' \quad \text{integers} \end{aligned}\right\} \quad \text{For all } i \text{ and } j$$

where

$I = \{1,2,\ldots,m\}$

$J = \{1,2,\ldots,n\}$

c_i = cost of part i

d_j = demand for application j

r_{ij}' = number of part i units required in application j.

x_i' = the number of part i on the module (decision variable)

y_j' = the number of modules needed in application j (decision variable)

In the previous approaches to this problem listed in the bibliography and also in the present paper, the problem is modified to the <u>continuous modular design problem</u> by dropping the integer requirements on x_i' and y_j' for all i and j; it is believed that a solution to the continuous problem is a necessary prelude to any solution of the integer version.

After the Evans paper, a second paper on modular design was written by A. Charnes and M. Kirby [2]. Both of these papers, proposed solution procedures based on searching the x-y space via specialized search routines. A third paper [6], by A. Passy, modified Charnes and Kirby's procedure by formulating the model as a geometric programming problem. The approach presented in Passy's paper is to move from one system of tight constraints to another until the optimum is found. Although Passy's procedure avoids the relatively slow search procedures of the first two papers, it has three major drawbacks. First, convergence of the procedure was not proved. (The results of the present paper might aid in proving convergence for Passy's procedure.) Second, procedures outlined by Passy to solve the problem of cycling could (and most likely, would) lead to an inordinate number of pivots. And, finally, the complexity of the calculations for each pivot step could require excessive amounts of computer time even if convergence were proved.

It is the intent of the present paper to:

1.) Develop a dual problem (with properties similar to the linear programming dual) from the Kuhn-Tucker optimality conditions.

2.) Use the dual solutions to develop a simplex-like solution algorithm and prove its convergence.

Preliminary tests indicate that the algorithm presented in this paper is very efficient. The authors have found that a continuous modular design problem can be solved with only slightly more computation time than a transportation problem of the same size. They are presently working on further computational tests of the procedure which will be reported on elsewhere.

2. DUALITY IN THE MODULAR DESIGN PROBLEM

Evans modified the original problem by making the following substitutions:

$$x_i = c_i \, x_i'$$

$$y_j = d_j\, y_j'$$

$$r_{ij} = r_{ij}'\, c_i\, d_j$$

He also noted that there exists an infinite number of solutions to the problem since if $\bar{x}$ and $\bar{y}$ are solution vectors, then so are $\bar{x}/\theta$ and $\bar{y}\cdot\theta$ for any $\theta > 0$. We may, therefore, add the restriction $\sum_j y_j = 1$ which singles out a unique member of the class without loss of generality. These transformations lead to the primal problem (P):

$$\min_x \sum_i x_i = g \tag{1P}$$

$$\text{s.t.} \quad x_i y_j - z_{ij} = r_{ij} \quad \text{for all } i \text{ and } j \tag{2P}$$

$$\sum_j y_j = 1 \tag{3P}$$

$$z_{ij},\ x_i,\ y_j,\ r_{ij} \geq 0 \quad \text{For all } i \text{ and } j \tag{4P}$$

where z_{ij} is a surplus variable.

<u>Theorem 1</u> (Evans) The solution to problem (P) exists and is unique.

This theorem was first proved by Evans [4]. The algorithm established in the present paper gives an alternate, constructive proof of the existence of a solution to problem (P).

LEMMA 1. If the requirements matrix, R, has no zero rows or columns, then in any solution, x_i, $y_j > 0$ for all i and j.

Proof. The assertion is obvious since $x_i y_j > 0$ at least once for each i and j.

Let λ_{ij} be associated with constraints (2P) and μ be associated with constraint (3P). Then it is easy to show that the Kuhn-Tucker conditions associated with the primal problem are:

$$\sum_j \lambda_{ij} y_j = 1 \quad \text{for all } i \epsilon I \tag{1C}$$

$$\sum_i x_i \lambda_{ij} = \mu \text{ for all } j \epsilon J \tag{2C}$$

$$z_{ij} \cdot \lambda_{ij} = 0 \qquad \text{For all } i \text{ and } j \tag{3C}$$

$$\lambda_{ij} \geq 0 \qquad \text{For all } i \text{ and } j \tag{4C}$$

LEMMA 2. For any pair of feasible solutions to the primal and dual problems, $\mu = g$.

PROOF. Multiply (1C) by x_i, sum over all i, and use (1P) to show:

$$\sum_i \sum_j x_i \lambda_{ij} y_j = \sum_i x_i = g$$

Now multiply (2C) by y_j, sum over all j and use (3P) to show:

$$\sum_i \sum_j x_i \lambda_{ij} y_j = \mu \sum_j y_j = \mu$$

Hence, $\mu = g$.

By analogy with classical linear programming we shall interpret constraints (3C) as "complementary slackness" conditions and insure that they hold by the algorithmic solution techniques we develop. Again by analogy to linear programming we add the objective function $\sum_i \sum_j \lambda_{ij} r_{ij}$ to constraints (1C), (2C), and (4C) to create the dual problem. The remainder of this section is devoted to showing that most of the simplex method solution techniques involving interplay between the primal and dual problems can be carried over to the modular design problem considered here. In later sections we show that they are powerful enough to make possible an efficient simplex-like algorithm for solving the continuous modular design problem.

The dual problem is defined by:

$$\underset{\lambda_{ij}}{\text{Maximize}} \quad \sum_i \sum_j \lambda_{ij} r_{ij} = f \tag{1D}$$

$$\text{s.t.} \quad \sum_j \lambda_{ij} y_j = 1 \quad \text{for all } i \tag{2D}$$

$$\sum_i x_i \lambda_{ij} = g \quad \text{for all } j \tag{3D}$$

$$\lambda_{ij} \geq 0 \quad \text{for all } i \text{ and } j \tag{4D}$$

LEMMA 3. For any pair of primal dual solutions (whether nonnegative or not): $g - f = \sum_i \sum_j \lambda_{ij} \cdot z_{ij}$.

PROOF. Multiply (2P) by λ_{ij}, sum over all i,j and use (1D) to show:

$$\sum_i \sum_j \lambda_{ij} x_i y_j - \sum_i \sum_j \lambda_{ij} z_{ij} = \sum_i \sum_j \lambda_{ij} r_{ij} = f$$

but $\sum_i \sum_j \lambda_{ij} x_i y_j = g$ (see proof of Lemma 2)

hence, $g - f = \sum_i \sum_j \lambda_{ij} z_{ij}$.

LEMMA 4. (Complementary Slackness.) For any pair of primal dual feasible solutions $g = f$ iff $\lambda_{ij} \cdot z_{ij} = 0$ for all i,j.

PROOF. The proof follows directly from Lemma 3 and the fact that λ_{ij} and z_{ij} are non-negative for all i and j.

THEOREM 2 (Duality Theorem). The quantities $\bar{x}_i$, $\bar{y}_j$ for all i and j are a solution to problem (P) iff $\bar{\lambda}_{ij}$ for all i and j are a solution to problem (D) and $g = f$.

PROOF. [The proof of this theorem will be only briefly sketched.] The Arrow-Hurwicz-Uzawa Constraint Qualification [5, p. 102] can be shown to hold for the constraint set of the primal problem. This implies that, at the optimum solution to the primal problem there exists a solution to the Kuhn-Tucker problem [5, pp. 105-106]. The functions defining the constraint set can be shown to be quasi-concave in the non-negative orthant while the objective function is linear. Zangwill [12, p. 43] has shown that in such a case a solution to the Kuhn-Tucker problem can occur only at

the optimum of the primal problem. Finally the K-T problem is made up of the constraint set of the dual problem together with the complementary slackness conditions $\lambda_{ij} \cdot z_{ij} = 0$ which are true if and only if $g = f$ (Lemma 4). This completes an outline for the proof of Theorem 2.

A property of interest, although not used here is that problems (P) and (D) are not mutually-dual. If we form the dual of problem (D) we get the following problem:

$$\min_{\zeta_i, \eta_j} \quad \sum_i \zeta_i + g \sum_j \eta_j = g' \qquad (1E)$$

$$\text{s.t.} \quad x_i \eta_j + \zeta_i y_j \geq r_{ij} \quad \text{for all } i \text{ and } j \qquad (2E)$$

where, ζ_i is the dual variable associated with constraint (2D) and η_j is the dual variable associated with constraint (3D).

For fixed x_i and y_j problems (D) and (E) are mutually dual generalized transportation problems. If $\bar{x}_i$ and $\bar{y}_j$ are solutions to (P) then $\bar{\zeta}_i = \bar{x}_i$, $\bar{\eta}_j = 0$ are a feasible solution to (E) from which it easily follows that $g \geq g'$.

3. THE RESTRICTED PROBLEM AND ITS SOLUTION

In this section it is first necessary to define some properties of graphs. For a general discussion of graph theory see Berge [1]. The following description was taken from [10, p.2].

"Let V be a set of n elements called <u>vertices</u> or <u>nodes</u> and let E be a set of (some of the) pairs (u,v) with $u,v \in V$. A pair (u,v) is called an <u>edge</u> between u and v, or also between v and u (no direction is implied). Then $G = (V,E)$ is called a <u>graph</u>. A <u>path</u> between u and v in G is a list:

$$u = w_0, w_1, \ldots, w_t = v$$

where, $(w_{j-1}, w_j) \in E$ for $j = 1,\ldots,t$. A path is a <u>cycle</u> if $u = v$ in the above list. A graph is <u>acyclic</u> if it has no cycles. A graph is <u>connected</u> if there is at least one path connecting each pair of distinct nodes. A <u>tree</u> is a connected acyclic graph. Equivalently, a graph is a tree if and only if there is a unique path between each pair of distinct nodes."

In addition to the definitions quoted above we shall need the following. A <u>forest</u> is an acyclic graph. It is easy to show that a forest is the union of trees, that is, a union of connected acyclic graphs.

In the modular design problem we shall consider the graph $G = (V,E)$ defined as follows: The set V of nodes consists of the rows and columns of the requirements matrix R; the set E of edges consists of some of the <u>cells</u> (i,j) of the R matrix.

Suppose G has a cycle

$$\Gamma = \{(s_1,t_1), (s_2,t_2),\ldots,(s_\ell,t_\ell)\}$$

where, $s_p = s_{p+1}$ or $t_p = t_{p+1}$ for $p = 1,2,\ldots,\ell$.

(Note $s_{\ell+1} = s_1$ and $t_{\ell+1} = t_1$) and ℓ is an even number. In each row or column of the R matrix there are either zero or two cells of the cycle. Then Γ can be written. $\Gamma = \Gamma_1 \cup \Gamma_2$ where:

$$\Gamma_1 = \{(s_1,t_1), (s_3,t_3),\ldots,(s_{\ell-1}, t_{\ell-1})\}$$

$$\Gamma_2 = \{(s_2,t_2), (s_4,t_4),\ldots,(s_\ell,t_\ell)\}$$

DEFINITION. The <u>value</u> $w_{(s,t)}$ of a cycle relative to <u>any</u> element $(s,t) \in \Gamma_2$ is defined to be the ratio

$$w_{(s,t)} = \frac{\prod_{(u,v)\in\Gamma_1} r_{u,v}}{\prod_{(k,p)\in\Gamma_2} r_{k,p}}$$

if all $r_{kp} > 0$ for $(k,p)\in\Gamma_2$; otherwise $w_{(s,t)} = \infty$.

DEFINITION. A <u>nondegenerate</u> problem is one having $w_{(s,t)} \neq 1$ for all cycles Γ and $(s,t) \in \Gamma_2$.

LEMMA 5. A problem with data r_{ij} for all i and j, may be replaced by a nondegenerate problem with perturbed data $r^*_{ij} = r_{ij} + \delta^{i+mj}$ if $r_{ij} \neq 0$ or $r^*_{ij} = 0$ if $r_{ij} = 0$ and where δ can be chosen arbitrarily small.

PROOF. For given m and n there are only a finite number of possible cycles Γ. For such a cycle to have value 1 with the perturbed data we must have

$$\prod_{(i,j)\in\Gamma_1} (r_{ij} + \delta^{i+mj}) - \prod_{(s,t)\in\Gamma_2} (r_{st} + \delta^{s+mt}) = 0.$$

This expression is a polynomial in δ. Moveover there is at least one power of δ that has a non zero coefficient. To show this, let (h,k) be the cell in Γ with smallest k, and given this k the smallest h; suppose $(h,k)\in\Gamma_1$ (a similar proof holds for $(h,k)\in\Gamma_2$). Then there is a term $c\delta^{h+mk}$ where $c = \prod_{(i,j)\in\Gamma_1-\{(h,k)\}} r_{ij} \neq 0$ since the value of the cycle is one and all other terms have higher powers of δ. Hence we need only choose δ small and not equal to any of a finite number of zeros of a finite number of polynomials to obtain a nondegenerate perturbed problem close to the original one.

In the rest of this paper, we shall assume that we are dealing with a nondegenerate problem. The techniques for extending the algorithm we shall present to degenerate problems are similar to those for linear programming and will not be discussed.

DEFINITION. Given a feasible solution to problem (P) by a <u>tight constraint</u> in row i we shall mean a cell (i,j) such that $x_i y_j - r_{ij} = z_{ij} = 0$.

LEMMA 6. At an optimum solution of problem (P) there is at least one tight constraint in each row and column of R.

PROOF. Assume the contrary, that we have an optimal solution and for some row u no cell is tight, i.e., $x_u y_j > r_{uj}$ for all j. But then we can decrease x_u while keeping the solution feasible and, therefore, decrease the objective function, which is a contradiction. By the problem symmetry, the same kind of proof is valid for columns of R.

LEMMA 7. If x_i, y_j, and z_{ij} are solutions to problem (P) then the graph with nodes being rows and columns of R and edges being the tight cells (i.e., $z_{ij} = 0$) is a forest.

PROOF. The proof follows directly from Lemma 6 and the fact that we are dealing with only non-degenerate problems, therefore eliminating the possibility of cycles.

We shall now define a forest-restricted problem associated with a given forest, F, to be:

$$\min_{x_i, y_j} \sum_{i \in I} x_i = g \tag{1H}$$

$$\text{s.t.} \quad \sum_{j \in J} y_j = 1 \tag{2H}$$

$$x_i y_j = r_{ij}, \quad (i,j) \in F \tag{3H}$$

$$x_i, y_j \geq 0 \quad \text{for all } i \text{ and } j \tag{4H}$$

A tree restricted problem is a forest-restricted problem where the forest is made up of a single tree. In the algorithm to be presented in this paper the authors retain a tree basis throughout the procedure. We are, therefore, primarily interested in solutions to tree-restricted problems. The solution to a tree-restricted problem will now be characterized.

For any tree-restricted solution there exists a unique path connecting

any two columns of R [1], [10]. This unique path may be presented as in Figure 1, for the case that column 1 is connected to column q. We have displayed only those columns and rows from R that correspond to the path between column q and column 1. In Figure 1 the cells where r's appear are all tight and rows and columns have been permuted and relabeled to be in the staircase form showed. Certain other cells may be tight in these rows and columns but are not of interest at this time, hence they are not indicated in Figure 1.

r_{11}	r_{12}				
	r_{22}	r_{23}			

			⋮		
			. $r_{p-1,q-1}$		
			$r_{p,q-1}$	r_{pq}	
				.	
				⋮	

Figure 1

It is easy to see that

$$y_q = \frac{r_{pq}}{r_{p,q-1}} \cdot y_{q-1} \qquad \text{since} \quad x_p y_q = r_{pq}$$

and $x_p y_{q-1} = r_{p,q-1}$. The procedure can be continued in a similar fashion until finally:

$$y_q = \frac{r_{pq}}{r_{p,q-1}} \cdot \frac{r_{p-1,q-1}}{r_{p-1,q-2}} \cdots \frac{r_{23}}{r_{22}} \cdot \frac{r_{12}}{r_{11}} \quad y_1 = d_{q1} y_1 \quad (1K)$$

In general denote by d_{ut} the ratio $\frac{y_u}{y_t}$. It will always be the case, as above, that d_{ut} is the quotient of products of r_{ij}'s. We can choose an arbitrary column, say column k, in the matrix R and represent all values of y_j for $j \in J$, in terms of y_k by means of the equation $y_j = d_{jk}\, y_k$. Note that $d_{kk} = 1$. Using constraint (2H) we have the solution to the tree restricted problem as $\sum_{j \in J} y_j = \sum_{j \in J} d_{jk} \cdot y_k = 1$ which yields

$$y_k = \frac{1}{\sum_{j \in J} d_{jk}}$$

and $y_v = d_{vk} y_k$ for all $v \in J$. The values of x_i are easily determined from constraint (3H).

Given a tree basis T, the associated restricted tree dual solution can be derived by using the fact that $\lambda_{ij} = 0$ for (i,j) not contained in the tree basis. This follows since these cells are not forced to be tight. Also, $z_{ij} = 0$ for (i,j) contained in the tree basis. Taking constraint (2P), multiplying by λ_{ij}/g and summing over i and j separately yields:

$$\sum_i \lambda_{ij}\, x_i\, y_j/g - \sum_i \lambda_{ij} \cdot z_{ij}/g = \sum_i \lambda_{ij} \cdot r_{ij}/g$$

and

$$\sum_j \lambda_{ij}\, x_i\, y_j/g - \sum_j \lambda_{ij} \cdot z_{ij}/g = \sum_j \lambda_{ij} \cdot r_{ij}/g$$

using constraints (2D) and (3D) and complementary slackness we may modify these equations to be:

$$y_j = \sum_{(i,j) \in T} \lambda_{ij} \cdot r_{ij}/g \qquad \text{for } j \in J \qquad (1L)$$

and

$$x_i = \sum_{(i,j) \in T} \lambda_{ij} \cdot r_{ij} \qquad \text{for } i \in I \qquad (2L)$$

It is interesting to note that the form of the equations for the variables λ_{ij} are the same as the form derived by Passy [6, p. 450] for the geometric dual

variables if we let $\rho_{ij} = \lambda_{ij} \cdot r_{ij}/g$. The dual objective function developed by Passy, however, is very different from the one used in this paper.

The tree structure of the nonzero variables in equations (1L) and (2L) means that the equations can be solved by a simple solution procedure. Note that they can be rewritten as

$$y_j = \sum_{(i,j)\in T} \rho_{ij} \quad \text{for } j\in J \qquad (1L')$$

$$x_i' = \sum_{(i,j)\in T} \rho_{ij} \quad \text{for } i\in I \qquad (2L')$$

where we have made the substitutions $x_i' = x_i/g$ and $\rho_{ij} = \lambda_{ij}\, r_{ij}/g$. The following algorithm finds the ρ_{ij}'s given the primal solution x_i', y_j:

(1) Let TR(TC) be the set of rows (columns) containing a unique tight cell. Because we have a tree-restricted solution $TR \cup TC$ is not empty.

(2) For all tight cells (i,j) with $i\in TR$ let $\rho_{ij} = x_i'$. For all tight cells (i,j) with $j\in TC$ let $\rho_{ij} = y_j$. That this is correct follows from the fact that these tight cells are unique in their rows or columns.

(3) "Cross out" the rows $i \in TR$ and $j \in TC$. For the remaining matrix define a new primal solution $\bar{x}_i'$ and $\bar{y}_j$ as follows

$$\bar{x}'_i = x_i' - \sum_{\substack{j\in TC \\ (i,j)\in T}} y_j \quad \text{for all } i \notin TR$$

$$\bar{y}_j = y_j - \sum_{\substack{i\in TR \\ (i,j)\in T}} x_i' \quad \text{for all } j \notin TC$$

(4) If there are no uncrossed out rows stop; otherwise go back to step (1) and repeat.

A few additional properties of tree-restricted solution will now be presented.

LEMMA 8. For any tree restricted solution let T be the tree basis and let (p,h) be a cell not in T. Then $T \cup \{(p,h)\}$ has a unique cycle whose value is

$$w_{(p,h)} = \frac{x_p y_h}{r_{ph}} .$$

PROOF. Let us assume (without loss of generality) that $h = 1$ and column 1 is connected to row p as in Figure 1. Adding the cell $(p,h) = (p,1)$ we obtain the cycle shown in Figure 2. We know that $y_q = d_{q1} y_1$ and

	1	2	. . . .		q		
1	r_{11}	r_{12}	....			....	
2	.	r_{22}	....				
.	.	.	....				
.				$r_{p-1,q-1}$			
p	$r_{p,1}$			$r_{p,q-1}$	$r_{p,q}$		
	.	.		.	.	.	

Figure 2

also that $x_p = r_{pq}/y_q$. Hence

$$x_p y_1 = \frac{r_{pq}}{d_{q1}}$$

where, as in (1K),

$$d_{q1} = \frac{r_{pq}}{r_{p,q-1}} \quad \frac{r_{p-1,q-1}}{r_{p-1,q-2}} \quad \cdots \quad \frac{r_{23}}{r_{22}} \cdot \frac{r_{12}}{r_{11}}$$

It follows that

$$\frac{x_p y_1}{r_{p1}} = \frac{r_{pq}}{r_{p1}} \; \frac{r_{p,q-1}}{r_{pq}} \; \frac{r_{p-1,q-2}}{r_{p-1,q-1}} \cdots \frac{r_{22}}{r_{23}} \; \frac{r_{11}}{r_{12}} = w_{(p,1)},$$

as was to be shown.

COROLLARY. A tree-restricted solution x_i, y_j is primal feasible if and only if every non basic cell determines a cycle whose value is ≥ 1.

This corollary is used to check for primal feasibility in the algorithm to be presented.

DEFINITION. Given a tree basis T and any cell $(p,q) \epsilon T$ we define the following four sets:

I_q = {set of all rows that can be reached in T using cell (p,q), except for row p}

$I_p = I - I_q$

J_p = {set of all columns that can be reached using cell (p,q), except for column q}

$J_q = J - J_p$

Clearly $p \in I_p$ and $q \in J_q$ and these sets are never empty. Also at most one of the sets I_q and J_p is empty.

	J_p	J_q
I_p		$\cdot\,(p,q)$
I_q		

Figure 3

Figure 3 shows the matrix R divided into the four subsets $I_p \times J_p$, $I_p \times J_q$, $I_q \times J_p$, and $I_q \times J_q$. Of these four sets; $I_q \times J_p$ contains no cells of T; $I_p \times J_q$ contains only the cell $(p,q) \in T$; all the rest of the cells of T are in the other two areas $(I_p \times J_p) \cup (I_q \times J_q)$.

In the algorithm to be presented later we are going to change data elements r_{pq} in a parametric fashion. The next two lemmas characterize what happens.

LEMMA 9. Let x_i, y_j be a primal feasible tree-restricted solution with tree basis T and let $(p,q)\epsilon T$. If we replace r_{pq} by a larger value $r^*_{pq} \geq r_{pq}$, then, provided

$$r^*_{pq} \leq \underset{(s,t)\epsilon I_q x J_p}{\text{Min}} \quad r_{pq} w_{(s,t)}$$

the tree-restricted solution x^*_i, y^*_j with the same basis, T, is primal feasible.

PROOF. By the corollary to Lemma 8 we need only show that every non basic cell (u,v) determines a cycle with value $w_{(u,v)} \geq 1$. Referring to Figure 3 it is obvious that non basic cells in the $(I_p \times J_p)$ or $(I_q \times J_q)$ areas have cycles entirely contained in these areas. Hence changing r_{pq} does not affect the values of their cycles.

Because every cycle goes alternately from row nodes to column nodes, every non-basic cell (u,v) in the $I_p \times J_q$ area determines a cycle Γ which includes (p,q) in the Γ_1 and (u,v) in the Γ_2 part. Hence $w_{(u,v)}$ increases if r_{pq} increases and primal feasibility continues to hold for these cells however large we make r^*_{pq}.

Finally consider cells (s,t) in the $I_q \times J_p$ area. Such cells determine a cycle Γ with (s,t) and (p,q) in the Γ_2 part. Let $w_{(s,t)}$ and $w^*_{(s,t)}$ be the value of the cycle determined by (s,t) with r_{pq} and $r^*_{pq} \geq r_{pq}$ respectively. Then $w^*_{(s,t)}\, r^*_{pq} = w_{(s,t)}\, r_{pq}$ by the definition of the value of a cycle. Since we want $w^*_{(s,t)} \geq 1$ this means that we must have $r^*_{pq} \leq w_{(s,t)}\, r_{pq}$ for every $(s,t) \in I_q \times J_p$ and therefore the statement of the Lemma is true.

LEMMA 10. Let x_i, y_j be a primal feasible tree-restricted solution with the tree basis T and let $(p,q)\epsilon T$. If we replace r^*_{pq} by a smaller value $r^{**}_{pq} \leq r^*_{pq}$, then provided

$$r_{pq}^{**} \geq \max_{(u,v)\in I_p \times J_q - \{(p,q)\}} \frac{r_{pq}^{*}}{w_{(u,v)}}$$

the tree-restricted solution x_i^*, y_j^* with the same basis, T, is primal feasible.

PROOF. The proof here is analogous to that for Lemma 9 and will not be given.

4. THE MODIFIED PROBLEM

Given problem R with data r_{ij} we define the <u>modified problem</u> R^* with data r_{ij}^* where $r_{ij} \leq r_{ij}^* < \infty$ for all i and j. We shall give an algorithm for finding an optimal tree-restricted solution to R^* and show how this can be used to find the optimal forest basis solution to Problem R.

THEOREM 3. Given an optimal tree-restricted solution to R^* there corresponds a unique forest-restricted optimal solution to R. Conversely, to the optimal forest-restricted solution to problem R there correspond at least one optimal tree-restricted solution to R^*.

PROOF. Given an optimal tree-restricted solution x_i^*, y_j^*, λ_{ij}^* with tree basis T to problem R^* drop from T all tight cells such that $\lambda_{ij}^* = 0$. The result is a forest F with $\lambda_{ij}^* > 0$ for $(i,j) \in F$ and $\lambda_{ij}^* = 0$ for $(i,j) \notin F$. Since $x_i^* y_j^* \geq r_{ij}^* \geq r_{ij}$ for all i and j these solutions are primal and dual feasible, and hence by the duality theorem are optimal for problem R.

Given an optimal forest-restricted solution x_i, y_j and λ_{ij} with forest basis F we shall give a constructive procedure for deriving an optimal forest restricted solution to a problem R^*. Suppose $F = T_1 \cup T_2 \cup \ldots \cup T_k$ where each T_i is a tree. If row u contains a (tight) cell of T_i then it will not contain a cell from any other tree. Similarly, if column v contains a cell of T_i then it will not contain a cell of any other tree. We now show one way to "hook together" the trees in F and make them into a single tree.

Let i_1 be the index of any row containing a cell of T_1 and let $j_2,\ldots,j_k$ be indices of columns containing cells of $T_2,\ldots,T_k$. Add the cells (i_1,j_2), $(i_1,j_3),\ldots,(i_1,j_k)$ to F which will make it into a connected tree and also define problem R^* by

$$r^*_{i,j_2} = x_{i_1} y_{j_2} \geq r_{i,j_2},\ldots,r^*_{i,j_k} = x_{i_1} y_{j_k} \geq r_{i_1 j_k}$$

and all other $r^*_{ij} = r_{ij}$. It follows that x_i, y_j and λ_{ij} are still primal and dual feasible and hence optimal for R^*.

Obviously, there are many other ways the trees $T_1, T_2,\ldots,T_k$ may be connected to make a single tree so the above process is not unique.

THEOREM 4. Let x_i, y_j be a feasible solution to problem R^* with tree basis T and let λ_{ij} be the restricted dual solution; and consider a cell $(p,q) \in T$; then

(a) if $\lambda_{pq} < 0$ we can decrease g^* by increasing r^*_{pq}

(b) if $\lambda_{pq} > 0$ we can decrease g^* by decreasing r^*_{pq}.

These results hold only over a sufficiently small range.

PROOF. Suppose we set $r^*_{ij} = r_{ij} + \delta_{ij}$ and write the Lagrangian function of the primal problem. It is

$$L(x,y,\lambda,\mu,z,\delta) = \sum_i x_i - \sum_i \sum_j \lambda_{ij}[x_i y_j + z_{ij} - (r_{ij} + \delta_{ij})]$$

$$+ \mu(\sum_j y_j - 1)$$

Now holding all $\delta_{u,v}$ fixed at zero except for δ_{ij} and letting $g(\delta_{ij})$ be the corresponding value of the primal problem we can rewrite the Lagrangian (for small changes in δ_{ij}) as

$$g(\delta_{ij}) = g(0) + \lambda_{ij} \delta_{ij} + o(\delta_{ij})$$

for which the two assertions are obvious.

THEOREM 5. Given a feasible tree-restricted solution x_i^*, y_j^*, λ_{ij}^* to problem R^* with basis T, let $(p,q) \in T$. Define r_{pq}^o to be the value of r_{pq}^* for which $\lambda_{pq}^* = 0$, then

$$r_{pq}^o = \sqrt{\frac{\sum\limits_{(i,j)\in R_p} \frac{r_{ij}^*}{e_{ip}}}{\sum\limits_{i\in I_p} e_{ip}} \cdot \frac{\sum\limits_{(i,j)\in C_q} \frac{r_{ij}^*}{d_{jq}}}{\sum\limits_{j\in J_q} d_{jq}}}$$

where the quantities e_{ip}, d_{jq}, R_p, and C_q will be explained in the proof below. Then

(a) $\lambda_{pq}^* = 0 \Leftrightarrow r_{pq}^* = r_{pq}^o$

(b) $\lambda_{pq}^* < 0 \Leftrightarrow r_{pq}^* < r_{pq}^o$

(c) $\lambda_{pq}^* > 0 \Leftrightarrow r_{pq}^* > r_{pq}^o$

PROOF. Let the sets I_p, I_q, J_p and J_q be as previously defined. If we remove (p,q) from T then as noted before all the remaining cells of T are in the areas $I_p \times J_p$ and $I_q \times J_q$.

For each $j \in J_p$ let i be the smallest row index such that $(i,j) \in T$; and let R_p be the set of such cells (i,j). Then $y_j = r_{ij}^*/x_i$ for $(i,j) \in R_p$ Also let e_{ip} be the ratio $e_{ip} = x_i/x_p$. Similarly for each i in I_q let j be the smallest column index such that $(i,j) \in T$; and let C_q be the set of all such cells. Then $x_i = r_{ij}^*/y_j$ for $(i,j) \in C_q$. As before let $d_{jq} = y_j/y_q$.

Removing (p,q) from T forces λ_{pq} to 0 and we can calculate the new solution x_i and y_j for all i and j. From this we can find r_{pq}^o as $x_p y_q$. Once (p,q) is dropped from T the primal problem (P) becomes

$$\text{Minimize} \sum_{i\in I_p} x_i + \sum_{i\in I_q} x_i = \sum_{i\in I_p} e_{ip}\, x_p + \sum_{(i,j)\in C_q} \left(\frac{r_{ij}}{d_{jq}}\right) \frac{1}{y_q}$$

subject to the constraint

$$\sum_{j \in J_r} y_j \; [illegible] \; \frac{1}{} + \sum_{j \in J_q} d_{jq} \Big) \, y_q = 1 \; .$$

Taking the second [illegible] in each case we see we have a constrained minimization [illegible] variables y_p and y_q. The standard Lagrange multiplier [illegible] unique solution

$$y_p = L \, / \, [illegible]$$

and

[illegible]

where L is a constant [illegible] rmine since we are only interested in the [illegible] these two expressions together gives r^{o}_{pq} [illegible] (a) follows.

For the [illegible] If $r^{*}_{pq} < 0$ we can decrease g by increasing [illegible] r^{*}_{pq} to get closer to 0, since otherwise [illegible] bound and continue to decrease [illegible] that [illegible] 0 [illegible]

r^{*}_{pq} [illegible] r^{o}_{pq} [illegible]

THEOREM [illegible]

to [illegible] with [illegible] we have

(a) r_{pq} [illegible] r^{o}_{pq}

(b) r_{pq} [illegible] r^{o}_{pq} [illegible] r_{pq}

(c) [illegible]

[illegible] that we could reduce [illegible] still further. [illegible]

(2) [illegible]

[illegible] we have a prime [illegible] feasible solution [illegible]

[illegible] condition [illegible] necessary [illegible] be [illegible] problem [illegible] there is no reason to [illegible] r_{pq} [illegible] not in the basis.

[illegible] THE ALGORITHM

[illegible] and R, [illegible]

(1) [illegible] to R^*. [illegible] shall later discuss two ways to find such [illegible]

(2) [illegible] from basis [illegible] to [illegible] method outlined in Section [illegible]

[illegible]

(iii) some cell (h,k) in $I_p \times J_q - \{(p,q)\}$ becomes tight; then add (h,k) to T remove (p,q) from T, set $r^*_{pq} = r_{pq}$, and go to 2.

(c) All cells $(p,q) \epsilon T$ satisfy $\lambda_{pq} \geq 0$ and $r^*_{pq} = r_{pq}$ or $\lambda_{pq} = 0$ and $r^*_{pq} = r^o_{pq} \geq r_{pq}$. Go to 4.

(4) The optimum tree-restricted solution to R^* is given by the current T, x_i, y_j and λ_{ij}, since these satisfy the duality theorem. The optimum forest F for R is obtained by dropping those cells (p,q) from T with $\lambda_{pq} = 0$; the same x_i, y_j and λ_{ij} are optimal for R with forest F.

THEOREM 7. For a non-degenerate problem the algorithm converges in an infinite number of steps to the optimum answer to problem R.

A proof of this theorem appears in Smeers [9].

An intuitive proof is the following. Suppose the algorithm always keeps the same tree; i.e., steps 2 (a) (ii) and 2 (b) (iii) are never entered. Then it is easy to show that the tree restricted problem (H), after making the substitutions $x_i = r_{ij}/y_j$ for $(i,j) \epsilon T$, is a concave problem, and the algorithm can easily be shown to converge (infinitely), see Zangwill [12]. Clearly there are only a finite number of possible tree structures, and each one has (in the non-degenerate case) a different optimum value. It follows that steps 2(a) (ii) and e(b) (iii) of the algorithm will be entered only a finite number of times so that the processes eventually settles down on a single tree for which the previous argument holds.

Since the above algorithm involves finding square roots at each step which are implemented by algorithms that converge infinitely, there can be no strictly finite procedure for solving the above problem. However, the following modification of step (2) of the algorithm ensures that all the rest of the calculations are finite.

(2′) Examine the cells (p,q) in the current basis T to find the set S of all cells (p,q) such that either (a) $\lambda_{pq} < 0$ or (b) $\lambda_{pq} > 0$ and $r^*_{pq} > r_{pq}$. If $S = \emptyset$ go to (4). Otherwise let $F = T - S$ and solve the resulting forest restricted problem (H) by a Lagrangian procedure similar to that in the proof of Theorem 5. If $x_p y_q > r_{pq}$ for all $(p,q) \in S$ then the optimum forest restricted solution has been found. If not, add cells of S to F in all possible ways and resolve until the condition holds. A branch and bound procedure can be devised to simplify this calculation.

It is easy to show that with step (2′) the algorithm becomes a finite one (except for the square roots) since there are only a finite number of forest bases, and step (2′) finds the minimum objective value for each one when it is considered. Hence no forest basis can be considered more than once in the computation process.

Since computers have limited accuracy, infinitely convergent processes have to be (finitely) terminated after they converge to that accuracy. The authors have found that the original algorithm with step (2) instead of (2′) converges quickly to within the accuracy limit determined by the computer. In fact, the number of pivot steps needed seems to be fewer than that needed for a corresponding transportation problem (see [11]). Therefore, step (2′) has not (as yet) been programmed.

In our code we have implemented degeneracy prevention techniques similar to those used in linear programming.

Smeers [9] has proposed an alternate way of finite termination for the algorithm.

We now discuss two ways of implementing the starting routine of the algorithm. The first procedure is similar to the improvement routine of the algorithm.

Starting Routine 1. Find an initial tree basis T by any means. A good heuristic is to try and get as many of the large entries in R as possible into this initial basis. Now solve for x_i and y_j using T. For each non basic cell (i,j) if $x_i y_j < r_{ij}$ replace r_{ij} by $r^*_{ij} = x_i y_j$. After this has been done go back and for each (i,j) such that $r^*_{ij} < r_{ij}$ increase r^*_{ij} until either a new cell becomes tight and enters the basis in place of (i,j) or else $r^*_{ij} = r_{ij}$. Note that this makes g constantly increase and hence these steps are just the reverse of the improvement routine of the algorithm. After a finite number of such steps a primal feasible tree-restricted solution to the original problem will be attained.

Starting Routine 2. Select an arbitrary set of positive y_j's such that $\Sigma y_j = 1$. A good choice would be to select

$$y_j = (\sum_i r_{ij})/(\sum_i \sum_j r_{ij})$$

Now choose $x_i = \operatorname{Max}_j (r_{ij}/y_j)$ and put all tight cells into the basis. If there is a column, say q. with no tight cells, select an arbitrary row, say p, and raise r^*_{pq} to the value $x_p y_q$ and add this cell to the basis also. We now have a primal feasible forest basis which can further be extended to a primal feasible tree basis by using the techniques of the proof of Theorem 3.

As a final remark, we would like to discuss how a primal feasible tree-solution can be used to determine the next tree solution. For the first solution we have $y_j = d_{jk} y_k$ for k fixed and all j. If we now change r^*_{pq} for (p,q) being a tight cell, the values of d_{jk} will be

affected only for those y_j such that (p,q) is a part of the unique path from column j to column k. If r^*_{pq} is changed to r^{**}_{pq} then either

$$d^*_{jk} = d_{jk} \frac{r^*_{pq}}{r^{**}_{pq}} \quad \text{or} \quad d^*_{jk} = d_{jk} \frac{r^{**}_{pq}}{r^*_{pq}}$$

depending on where (p,q) is in the path from column j to column k.

6. EXAMPLES

The first example is designed to demonstrate most of the steps of the algorithm. We start with the data and an initial tree as

5	5	4
3	10	7
4	3	4

Cells (3,3) and (1,3) are not primal feasible since

$$4 > \frac{7 \cdot 5 \cdot 4}{10 \cdot 5} = 2.8 \quad \text{and} \quad 4 > \frac{5 \cdot 7}{10} = 3.5$$

Following Starting Routine 1 we replace the problem by

5	5	3.5
3	10	7
4	3	2.8

whose value is $g = 51.3$. We now bring cell (3,3) into the basis and can remove any cell in $\Gamma_1 = \{(2,3), (1,2), (3,1)\}$. We choose to remove cell (3,1) and now increase $r^*_{3,3}$ from 2.8, trying to raise it to 4, without causing primal infeasibilities. We succeed and obtain the problem:

5	5	3.5
3	10	7
4	3	4

whose value is $g = 55.93$. We now bring cell (1,3) into the basis and can remove either cell in $\Gamma_1 = \{(1,2), (2,3)\}$ we choose to remove (2,3) (although later it will become evident that the other choice is better) in order to demonstrate more steps of the algorithm. We raise r^*_{13} to 4 without encountering primal infeasibilities and obtain the problem

5	5	4
3	10	7
4	3	4

whose value is $g = 56$. Its primal feasible solutions are

$$y = \left(\frac{5}{14}, \frac{5}{14}, \frac{4}{14}\right)$$

$$x = \left(14, 28, 14\right)$$

$$x' = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)$$

and the dual solutions are given by

$$\rho_{11} = y_1 = \frac{5}{14}, \quad \rho_{22} = x_2' = \frac{1}{2}, \quad \rho_{33} = x_3' = \frac{1}{4}$$

$$\rho_{12} = y_2 - \rho_{22} = \frac{5}{14} - \frac{1}{2} = -\frac{2}{14}, \quad \rho_{13} = y_3 - \rho_{33} = \frac{4}{14} - \frac{1}{4} = \frac{1}{28}$$

Since (1,2) is the only cell with negative dual variable we have $p = 1$, $q = 2$, $I_1 = \{2\}$, $J_2 = \{1,3\}$, $R_2 = \{(1,1),(3,3)\}$, $C_1 = \{(2,2)\}$ and $e_{11} = e_{13} = 1$, $d_{22} = 1$. Hence we can calculate r^o_{12} from Theorem 5 as

$$r^o_{12} = \sqrt{\frac{\frac{5}{1} + \frac{4}{1}}{2} \quad x \quad \frac{\frac{10}{1}}{1}} = \sqrt{45} = 6.71$$

We also have $I_1 \times J_2 = \{(2,1), (2,3)\}$ as the two cells that may become tight as we increase r_{12}. For them we have:

$$r^*_{12} \le \frac{5\cdot 10}{3} = 16.67 \qquad \text{from cell } (2,1)$$

and

$$r^*_{12} \le \frac{10\cdot 4}{7} = 5.71 \qquad \text{from cell } (2,3).$$

The smallest constraint comes from cell (2,3) so we bring it into the basis obtaining the problem:

5	5	4
3	10	7
4	3	4

whose value is $g = 55.18$. Its solution is

$$y = \left(\frac{35}{103}, \frac{40}{103}, \frac{28}{103} \right)$$

$$x = \left(\frac{103}{7}, \frac{103}{4}, \frac{103}{7} \right)$$

$$x' = \left(\frac{4}{15}, \frac{7}{15}, \frac{4}{15} \right)$$

The dual solution is given by:

$$\rho_{11} = y_1 = \frac{35}{103}, \quad \rho_{22} = y_2 = \frac{40}{103}, \quad \rho_{33} = x'_3 = \frac{4}{15}$$

$$\rho_{23} = x'_2 - \rho_{22} = \frac{7}{15} - \frac{40}{103} = .08, \quad \rho_{13} = x'_1 - \rho_{11} = \frac{4}{15} - \frac{35}{103} = -.07$$

Cell (1,3) is the only one with negative dual variable so $p = 1$, $q = 3$, $I_1 = \{2,3\}$, $J_3 = \{1\}$, $R_3 = \{(1,1)\}$, $C_1 = \{(2,2), (3,3)\}$, and $e_{11} = 1$, $d_{23} = \frac{10}{7}$, $d_{33} = 1$.

Hence we calculate

$$r^o_{13} = \sqrt{5 \times \frac{10.\frac{7}{10} + \frac{4}{1}}{1 + \frac{10}{7}}} = 4.76$$

We also have $I_1 \times J_3 = \{(2,1), (3,1)\}$ so that the other constraints on r^*_{13} are given by

$$r^*_{13} \leq \frac{5 \cdot 7}{3} = 11.67 \qquad \text{from cell } (2,1)$$

$$r^*_{13} \leq \frac{4 \cdot 5}{4} = 5 \qquad \text{from cell } (3,1)$$

Therefore we increase r^*_{13} to 4.76 obtaining the problem

5	5	4.76
3	10	7
4	3	4

whose value is $g = 54.83$. The primal and dual solutions are:

$$y = (.29,\ .41,\ .30)$$

$$x = (16.56,\ 24.35,\ 13.92)$$

$$x' = (.30,\ .44,\ .26)$$

$$\rho_{11} = y_1 = .30, \quad \rho_{13} = x_1' - \rho_{11} = 0, \quad \rho_{22} = y_2 = .41$$

$$\rho_{23} = x_2' - \rho_{22} = .44 - .41 = .03, \quad \rho_{31} = x_3' = .26.$$

Since all cells satisfy 3(c) of the algorithm, we have the optimum solution to both R and R^*. We next solve Evans' problem [4], which is given with an initial tree basis:

15	23	44
13	13	0
15	17	35
34	12	22

cells (2,2) and (4,2) are not primal feasible since

$$13 > \frac{13 \cdot 22 \cdot 23}{34 \cdot 44} = 4.4$$

$$12 > \frac{23 \cdot 22}{44} = 11.5$$

The modified primal-feasible problem is:

15	23	44
13	4.4	0
15	17	35
34	11.5	22

We now bring cell (2,2) into the basis and remove cell (2,1). The problem remains primal feasible and we can increase the value of r^*_{22} to 13 while remaining primal feasible.

15	23	44
13	13	0
15	17	35
34	11.5	22

$11.5 = \frac{23 \cdot 22}{44}$ so we bring cell (4,2) into the basis and remove cell (4,3). We then increase r^*_{42} to 12 remaining primal feasible.

Initial primal feasible solution to the original problem:

15	23	44
13	13	0
15	17	35
34	12	22

$y = (.49, .17, .33)$

$x = (132.17, 74.70, 105.13, 68.96)$

$x' = (.35, .20, .28, .18)$

Dual variables

$$\rho_{41} = y_1 = .49$$

$$\rho_{42} = x_4' - \rho_{41} = .18 - .49 = -.31$$

$$\rho_{33} = x_3' = .28$$

$$\rho_{22} = x_2' = .20$$

$$\rho_{13} = y_3' - \rho_{33} = .33 - .28 = .05$$

$$\rho_{12} = x_1' - \rho_{13} = .35 - .05 = .30$$

Only cell (4,2) has a negative dual variable.

Take cell (4,2): $I_q \times J_p = \{(1,1),\ (2,1),\ (3,1)\}$

For (1,1) $\quad 15 \leq \dfrac{23 \cdot 34}{r_{42}}$ so $r_{42}^* \leq 52.1$

For (2,1) $\quad 13 \leq \dfrac{13 \cdot 34}{r_{42}}$ so $r_{42}^* \leq 36.8$

For (3,1) $\quad 15 \leq \dfrac{35 \cdot 23 \cdot 34}{44 \cdot r_{42}}$ so $r_{42}^* \leq 41.4$

Therefore $r_{42}^* \leq 36.8$ but $r_{42}^o = 25.18$

Therefore we can raise r_{42} to 25.18

15	23	44
13	13	0
15	17	35
34	25.18	22

$y = (.32,\ .23,\ .45)$

$x = (98.00,\ 55.43,\ 78.01,\ 107.33)$

$x' = (.29,\ .16,\ .23,\ .32)$

$g = 338.77$

Dual variables:

$$\rho_{41} = y_1 = .32$$

$$\rho_{42} = x_4' - \rho_{41} = 0.0$$

$$\rho_{33} = x_3' = .23$$

$$\rho_{22} = x_2' = .16$$

$$\rho_{13} = y_3 - \rho_{33} = .45 - .23 = .22$$

$$\rho_{12} = x_1' - \rho_{13} = .29 - .22 = .07$$

Since we satisfy 3(c) of the algorithm, we have the optimal solution to R and R^*.

7. CONCLUSIONS

The algorithm presented in this paper has several advantages.

a.) If we must stop before the optimum solution to the problem is reached, the non-linearity makes g close to the optimum. This solution will be much closer than would be the case in a corresponding similar linear problem.

b.) The method always keeps a primal feasible solution so that one can stop the procedure at any time and have a usable solution.

c.) In order to find the negative restricted dual variable, only the basis tree must be searched. A most negative indicator rule would, therefore, be available at low computational cost. (This is not the case for transportation problem (see [11]).

d.) The search for new limiting cells requires searching only the areas $I_p \times J_q$ or $I_q \times J_r$, and not the whole matrix.

e.) Previous solutions to the primal problem can be used to generate feasible solutions to problems with similar data.

f.) Accuracy is not a problem since a solution to a forest basis can be found independently of any previous operations.

Retention of a tree solution throughout the computation and use of previous solutions makes the steps in the algorithm very similar to those of transportation problems. Srinivasan and Thompson report excellent computational results of 175 x 175 transportation problems in seven seconds [11]. The number of pivots required for the modular design problems tested so far are somewhat fewer than for a transportation problem of the same size. This means that modular design problems can be solved in only slightly more computer time than that required for comparable transportation problems. The authors are currently preparing a report on computational experience with their modular design code.

REFERENCES

[1] Berge, C., The Theory of Graphs and Its Applications, John Wiley and Sons, Inc., New York, N. Y., 1962.

[2] Charnes, A., and M. Kirby, "Modular Design, Generalized Inverses and Convex Programming," Operations Research, Vol. 13, pp. 836-847 (1965).

[3] Duffin, J., E. Peterson, and C. Zener, Geometric Programming, John Wiley and Sons, Inc., 1967.

[4] Evans, D., "Modular Design -- Special Case in Nonlinear Programming," Operations Research, Vol. 11, pp. 637-647 (1963).

[5] Mangasarian, O., Nonlinear Programming, McGraw-Hill Book Co., 1969.

[6] Passy, A., "Modular Design: An Application of Structured Geometric Programming," Operations Research, Vol. 18, pp. 441-453 (1970).

[7] Rutenberg, D. and T. Shaftel, "Product Design: Subassemblies for Multiple Markets," Management Science, forthcoming.

[8] Shaftel, T., "An Integer Approach to Modular Design," Operations Research, Vol. 19, pp. 130-134 (1971).

[9] Smeers, Y., "Geometric Programming with Applications to Management Science," Ph.D. Thesis, Carnegie-Mellon University, 1972.

[10] Srinivasan, V. and G. Thompson, "Accelerated Algorithms for Labelling and Relabelling Trees," Management Science Research Report No. 228, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1970.

[11] ______________ and ___________, "Benefit-Cost Analysis of Coding Techniques for the Primal Transportation Algorithm," Management Science Research Report No. 229, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1970.

[12] Zangwill, W., Nonlinear Programming A Unified Approach, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1969.